ELSEVIER

Discrete Applied Mathematics 109 (2001) 197–213

DISCRETE APPLIED MATHEMATICS

Best reduction of the quadratic semi-assignment problem

Alain Billionnet[a, *], Sourour Elloumi[b]

[a] CEDRIC, Institut d'Informatique d'Entreprise, 18 allée Jean Rostand, 91025 Evry Cedex, France
[b] CEDRIC, Conservatoire National des Arts et Métiers, 292 rue Saint-Martin, 75141 Paris Cedex 03, France

Received 13 September 1996; revised 4 April 2000; accepted 10 April 2000

Abstract

We consider the quadratic semi-assignment problem in which we minimize a quadratic pseudo-Boolean function F subject to the semi-assignment constraints. We propose in this paper a linear programming method to obtain the best reduction of this problem, i.e. to compute the greatest constant c such that F is equal to c plus F' for all feasible solutions, F' being a quadratic pseudo-Boolean function with nonnegative coefficients. Thus constant c can be viewed as a generalization of the height of an unconstrained quadratic 0–1 function introduced in (Hammer et al., Math. Program. 28 (1984) 121–155), to constrained quadratic 0–1 optimization. Finally, computational experiments proving the practical usefulness of this reduction are reported.

Keywords: 0–1 quadratic programming; Semi-assignment problem; Reduction; Linear programming; Roof duality

1. Introduction

The quadratic semi-assignment problem can be stated as

$$\min\ F(x) = \sum_{i=1,m} \sum_{k=1,n} e_{ik} x_{ik} + \sum_{i=1,m-1} \sum_{j=i+1,m} \sum_{k=1,n} \sum_{\ell=1,n} c_{ikj\ell} x_{ik} x_{j\ell}$$

$$\text{(QSAP) s.t.} \quad \sum_{k=1,n} x_{ik} = 1 \quad (i = 1,\dots,m),$$

$$x_{ik} \in \{0,1\} \quad (i = 1,\dots,m,\ k = 1,\dots,n),$$

where e_{ik} $(i = 1,\dots,m, k = 1,\dots,n)$ and $c_{ikj\ell}$ $(i = 1,\dots,m-1,\ j = i+1,\dots,m,\ k = 1,\dots,n,\ \ell = 1,\dots,n)$ are real coefficients. Let $g_i(x) = \sum_{k=1,n} x_{ik} - 1$ and $A = \{x\colon x \in \{0,1\}^{m\times n},\ g_i(x) = 0,\ i = 1,\dots,m\}$. A is the set of all feasible solutions of (QSAP).

* Corresponding author.
E-mail address: billionnet@iie.cnam.fr (A. Billionnet).

0166-218X/01/$ - see front matter © 2001 Elsevier Science B.V. All rights reserved.
PII: S0166-218X(00)00257-2

This problem belongs to the class of the NP-hard problems for which no polynomial-time algorithms are known. Some task-assignment problems in distributed systems can be easily formulated as quadratic semi-assignment problems (see, e.g. [4,7,8,16]). Another application of (QSAP) is studied in [10].

Definition 1 (*Reduction*)**.** Given an objective function $F(x)$ of (QSAP), we shall call a new objective function $F'(x)$ a *reduction* if

$$F(x) = c + F'(x) \quad \text{for all } x \in A, \tag{1}$$

where c is a constant and

$$F'(x) = \sum_{i=1,m} \sum_{k=1,n} e'_{ik} x_{ik} + \sum_{i=1,m-1} \sum_{j=i+1,m} \sum_{k=1,n} \sum_{\ell=1,n} c'_{ikj\ell} x_{ik} x_{j\ell}$$

with e'_{ik} $(i = 1,\dots,m,\ k = 1,\dots,n)$ and $c'_{ikj\ell}$ $(i = 1,\dots,m-1,\ j = i+1,\dots,m,\ k = 1,\dots,n,\ \ell = 1,\dots,n)$ real and nonnegative.

A reduction is interesting because it can always constitute an initial step in the resolution of a problem and can provide a substantial speed up of the computational time when resolving the original problem or even when computing only a feasible solution or a lower bound to it. It is very easy to find a reduction for (QSAP) and we present in Section 5.2 a few different straightforward reductions. However, it is more difficult to find the *best reduction* for a given criteria.

Definition 2 (*Best reduction*)**.** We shall call best reduction a reduction which corresponds to the greatest constant c and an objective function F' such that (1) is satisfied. In other words, if there exist a constant $c' > c$ and a pseudo-Boolean function F'' with nonnegative coefficients such that $F(x) = c' + F''(x)$ for all $x \in A$ then F'' is at least cubic.

In order to find and prove the best reduction, we will consider the general problem of rewriting the function F as a constant plus a quadratic posiform (i.e. a quadratic function of x and $\bar{x}$ with nonnegative coefficients). We recall in Section 2 a general result, presented in [5], for computing, given a quadratic pseudo-Boolean function $(q.p.B.f.)\, f(x)$, the greatest constant c such that $f(x) = c + \phi(x,\bar{x})$ for all $x \in X$, X being an arbitrary subset of $\{0,1\}^n$ and ϕ being a quadratic posiform. This result shows that, in order to compute c, one must characterize the set of q.p.B.f.'s which are equal to 0 for all $x \in X$. In Section 3 we characterize the unique polynomial form of the q.p.B.f.'s which are equal to 0 for all $x \in A$. In Section 4 we give the linear program (LP) whose optimal value is the greatest constant c and show that the obtained posiform is the best reduction for the objective function of (QSAP). In Section 5, we first discuss previous uses of the linearization technique yielding to the linear problem (LP) whose resolution gives the best reduction; then, using an example, we show that the best reduction is moreover strictly better than other classical LP-based reductions, and, finally, we report some experimental results which prove the practical usefulness of the best reduction. Using the most classical linearization method of a quadratic 0–1 function [15] and a MIP solver, finding an optimal solution is 2–50 times quicker when the best reduction method is integrated in the presolve process.

2. Computation of the greatest constant c such that $f(x) = c + \phi(x, \bar{x})$ for all $x \in X \subset \{0, 1\}^n$, f being a q.p.B.f. and ϕ a quadratic posiform

The unconstrained quadratic 0–1 minimization problem consists of determining the minimum over $\{0,1\}^n$ of a q.p.B.f.:

$$f(x) = q_0 + \sum_{i=1,n} q_i x_i + \sum_{i=1,n-1} \sum_{j=i+1,n} q_{ij} x_i x_j, \tag{2}$$

where q_0, q_i, and q_{ij} are arbitrary real numbers.

Hammer et al. [13] introduced the *height* of function f (notation: $H(f)$) as the greatest constant c such that there exist a quadratic posiform $\phi(x,\bar{x})$ such that:

$$f(x) = c + \phi(x,\bar{x}) \quad \text{for all } x \in \{0,1\}^n. \tag{3}$$

Moreover, several practical methods were proposed [1,11] to compute the height of a given q.p.B.f. For example, it is proved in [11] that $H(f)$ is equal to the optimal value of the following linear program:

$$\begin{aligned} \min \quad & q_0 + \sum_{i=1,n} q_i x_i + \sum_{i=1,n-1} \sum_{j=i+1,n} q_{ij} y_{ij} \\ \text{s. t.} \quad & y_{ij} \geqslant 0, \quad y_{ij} \leqslant x_i, \quad y_{ij} \leqslant x_j, \quad 1 - x_i - x_j + y_{ij} \geqslant 0 \\ & (i = 1, \ldots, n-1, \ j = i+1, \ldots, n) \end{aligned}$$

and the posiform associated to $H(f)$ can be easily deduced from the optimal solution of the linear program.

A generalization of the notion of height to the constrained quadratic 0–1 minimization problem is presented in [5]. The constrained quadratic 0–1 minimization problem consists of determining the minimum over X of a q.p.B.f. f, where X is an arbitrary subset of $\{0,1\}^n$. The objective in this case is to find a constant c and a quadratic posiform $\phi(x,\bar{x})$ such that:

$$f(x) = c + \phi(x,\bar{x}) \quad \text{for all } x \in X \tag{4}$$

and, the greatest constant c satisfying (4) is denoted by $H_X[f(x)]$. Following this notation, $H_{\{0,1\}^n}[f(x)]$ is the height of f and it is straightforward that $H_X[f(x)] \geqslant H_{\{0,1\}^n}[f(x)]$.

Together with this generalization, the authors of [5] proved the lemma we recall hereafter. This lemma describes a theoretical scheme for the computation of $H_X[f(x)]$ for an arbitrary q.p.B.f. f and an arbitrary subset X of $\{0,1\}^n$. According to this scheme, one needs to characterize the set of q.p.B.f. which are equal to 0 for all $x \in X$. Finally, in [5] is given a practical method to compute $H_X[f(x)]$ and the associated posiform when $X = \{x \in \{0,1\}^n, \sum_{i=1,n} x_i = k\}$. In the sequel, we present a practical method to compute $H_A[F(x)]$ and the associated posiform and show that this posiform is the best reduction for F.

Let us now recall the lemma in [5] and its proof.

Lemma 1 (Billionnet and Faye [5]). *Let f be a q.p.B.f., X be an arbitrary subset of $\{0,1\}^n$, and $F_0(n,X)$ be the set of q.p.B.f. which are equal to 0 for all $x \in X$.*

Then,

$$H_X[f(x)] = \max\{H(f+\gamma)\colon \gamma \in F_0(n,X)\}.$$

Proof. Let $c = H(f+\sigma) = \max\{H(f+\gamma)\colon \gamma \in F_0(n,X)\}$. Because $c = H(f+\sigma)$, there exist a quadratic posiform Γ such that $f(x)+\sigma(x) = c + \Gamma(x,\bar{x})$ for all $x \in \{0,1\}^n$ and because $\sigma \in F_0(n,X)$, $f(x) = c + \Gamma(x,\bar{x})$ for all $x \in X$. Hence, $H_X[f(x)] \geqslant c$.

Now, let $c' = H_X[f(x)]$. There exist ϕ', a quadratic posiform, such that $f(x) = c' + \phi'(x,\bar{x})$ for all $x \in X$. Let $\psi(x)$ be the unique polynomial form of the quadratic posiform $\phi'(x,\bar{x})$. Obviously $f(x) = c' + \psi(x)$ for all $x \in X$. Consider the q.p.B.f. $\sigma(x) = \psi(x) - f(x) + c'$. It is clear that this function belongs to $F_0(n,X)$ and $f(x) + \sigma(x) = c' + \psi(x) = c' + \phi'(x,\bar{x})$ for all $x \in \{0,1\}^n$. Hence, $c' \leqslant \max\{H(f+\gamma)\colon \gamma \in F_0(n,X)\}$. $\square$

3. Characterization of *a* q.p.B.f. equal to 0 for all $x \in A$

Theorem 1. *Let*

$$\Gamma(x) = q_0 + \sum_{i=1,m}\sum_{k=1,n} q_{ik}\, x_{ik} + \sum_{i=1,m-1}\sum_{j=i+1,m}\sum_{k=1,n}\sum_{\ell=1,n} q_{ikj\ell}\, x_{ik}\, x_{j\ell}$$
$$+ \sum_{i=1,m}\sum_{k=1,n-1}\sum_{\ell=k+1,n} \mu_{ik\ell}\, x_{ik}\, x_{i\ell};$$

then $\Gamma(x) = 0$ for all x in A (i.e. g is in F_0 $(n.m,A)$) if and only if $\exists\lambda_i \in \mathbb{R}$ $(i = 1,\ldots,m)$; $\exists\lambda_{ijk} \in \mathbb{R}$ $(i = 1,\ldots,m;\ j = 1,\ldots,m;\ i \neq j;\ k = 1,\ldots,n)$ such that

$$\Gamma(x) = \sum_{i=1,m} \lambda_i\, g_i(x) + \sum_{i=1,m}\sum_{j=1,m;j\neq i}\sum_{k=1,n} \lambda_{ijk}\, x_{jk}\, g_i(x)$$
$$+ \sum_{i=1,m}\sum_{k=1,n-1}\sum_{\ell=k+1,n} \mu_{ik\ell}\, x_{ik}\, x_{i\ell}. \tag{5}$$

Proof.

- The sufficient condition is obvious since $\forall x \in A$, $g_i(x) = 0$ $(i = 1,\ldots,m)$ and then $\forall k$, $\forall \ell \neq k$, $x_{ik}\, x_{i\ell} = 0$.

- Let us prove the necessary condition (notation : if $j < i$ then $\sum_{s=i,j} h_s = 0$). Let $x^0 \in \{0,1\}^{m\times n}$ be such that $(x^0)_{i1}=1$ $(i=1,\dots,m)$ and $(x^0)_{ik}=0$ $(i=1,\dots,m;\ k=2,\dots,n)$. x^0 is in A so $\Gamma(x^0)=0$ and therefore,

$$q_0 + \sum_{r=1,m} q_{r1} + \sum_{r=1,m-1}\ \sum_{s=r+1,m} q_{r1s1} = 0 \quad \text{(E)}.$$

For $i=(1,\dots,m)$ and $k=(1,\dots,n)$, let $y^{ik} \in \{0,1\}^{m\times n}$ be such that $(y^{ik})_{ik}=1$, $(y^{ik})_{ip}=0$ $(p=1,\dots,n;\ p\neq k)$, $(y^{ik})_{r1}=1$ $(r=1,\dots,m;\ r\neq i)$, and $(y^{ik})_{rp}=0$ $(r=1,\dots,m;\ r\neq i;\ p=2,\dots,n)$. y^{ik} is obtained from x^0 by switching the values of variables x_{ik} and x_{i1}. y^{ik} is in A so $\Gamma(y^{ik})=0$ and therefore,

$$q_0 + q_{ik} + \sum_{r=1,m;r\neq i} q_{r1} + \sum_{r=1,i-1} q_{r1ik} + \sum_{r=i+1,m} q_{ikr1}$$

$$+ \sum_{r=1,m-1;r\neq i}\ \sum_{s=r+1,m;s\neq i} q_{r1s1} = 0 \quad (\mathrm{E}(i,k)).$$

For $i=(1,\dots,m-1)$, $k=(1,\dots,n)$, $j=(i+1,\dots,m)$ and $\ell=(1,\dots,n)$, let $z^{ikj\ell} \in \{0,1\}^{m\times n}$ be such that $(z^{ikj\ell})_{ik}=1$, $(z^{ikj\ell})_{ip}=0$ $(p=1,\dots,n;\ p\neq k)$, $(z^{ikj\ell})_{j\ell}=1$, $(z^{ikj\ell})_{jp}=0$ $(p=1,\dots,n;\ p\neq \ell)$, $(z^{ikj\ell})_{r1}=1$ $(r=1,\dots,m;\ r\neq i;\ r\neq j)$, and $(z^{ikj\ell})_{rp}=0$ $(r=1,\dots,m;\ r\neq i;\ r\neq j;\ p=2,\dots,n)$. $z^{ikj\ell}$ is obtained from x^0 by switching the values of variables x_{ik} and x_{i1} on the one hand and those of variables $x_{j\ell}$ and x_{j1} on the other. $z^{ikj\ell}$ is in A so $\Gamma(z^{ikj\ell})=0$ and therefore,

$$q_0 + q_{ik} + q_{j\ell} + \sum_{r=1,m;r\neq i;r\neq j} q_{r1} + q_{ikj\ell} + \sum_{r=1,i-1} q_{r1ik} + \sum_{r=i+1,m;r\neq j} q_{ikr1}$$

$$+ \sum_{r=1,j-1;r\neq i} q_{r1j\ell} + \sum_{r=j+1,m} q_{j\ell r1}$$

$$+ \sum_{r=1,m-1;r\neq i;r\neq j}\ \sum_{s=r+1,m;s\neq i;s\neq j} q_{r1s1} = 0 \quad (\mathrm{E}(i,j,k,\ell)).$$

By computing "(E)–(E(i,k))", one can write, for $i=1,\dots,m$, and for $k=1,\dots,n$,

$$q_{ik} = q_{i1} - \sum_{r=1,i-1} q_{r1ik} - \sum_{r=i+1,m} q_{ikr1} + \sum_{r=1,i-1} q_{r1i1} + \sum_{r=i+1,m} q_{i1r1} \quad (\mathrm{E}'(i,k))$$

and by computing "$(\mathrm{E}(i,j,k,\ell)) + (\mathrm{E}) - (\mathrm{E}(i,k)) - (\mathrm{E}(j,\ell))$" one can write for $i=1,\dots,m-1$, for $j=i+1,\dots,m$, for $k=1,\dots,n$, and for $\ell=1,\dots,n$:

$$q_{ikj\ell} = q_{ikj1} + q_{i1j\ell} - q_{i1j1} \quad (\mathrm{E}'(i,j,k,\ell)).$$

Now, we are going to prove that the following values of λ_i $(i=1,\dots,m)$ and λ_{ijk} $(i=1,\dots,m;\ j=1,\dots,m;\ i\neq j;\ k=1,\dots,n)$ satisfy (5). Let

$$\lambda_i = q_{i1} + \sum_{j=i+1,m} q_{i1j1} \quad \text{for } i=1,\dots,m,$$

$$\lambda_{jik} = q_{ikj1} \quad \text{and} \quad \lambda_{ijk} = q_{i1jk} - q_{i1j1}$$

$$\text{for } i=1,\dots,m-1,\ j=i+1,\dots,m,\ k=1,\dots,n.$$

We easily obtain that (E) can be written as $q_0 = -\sum_{i=1,m} \lambda_i$, $(\mathrm{E}'(i,k))$ can be written as

$$q_{ik} = \lambda_i - \sum_{j=1,i-1} \lambda_{jik} - \sum_{j=i+1,m} \lambda_{jik} \quad (i = 1,\dots,m;\ k = 1,\dots,n)$$

and $(\mathrm{E}'(i,j,k,\ell))$ can be written as

$$q_{ikj\ell} = \lambda_{jik} + \lambda_{ij\ell} \quad (i = 1,\dots,m-1,\ j = i+1,\dots,m,\ k = 1,\dots,n,\ \ell = 1,\dots,n).$$

Rewriting $\Gamma(x)$, we get

$$\begin{aligned}
\Gamma(x) = {} & q_0 + \sum_{i=1,m} \sum_{k=1,n} q_{ik} x_{ik} + \sum_{i=1,m-1} \sum_{j=i+1,m} \sum_{k=1,n} \sum_{\ell=1,n} q_{ikj\ell} x_{ik} x_{j\ell} \\
& + \sum_{i=1,m} \sum_{k=1,n-1} \sum_{\ell=k+1,n} \mu_{ik\ell} x_{ik} x_{i\ell} \\
= {} & -\sum_{i=1,m} \lambda_i + \sum_{i=1,m} \sum_{k=1,n} \left(\lambda_i - \sum_{j=1,i-1} \lambda_{jik} - \sum_{j=i+1,m} \lambda_{jik} \right) x_{ik} \\
& + \sum_{i=1,m-1} \sum_{j=i+1,m} \sum_{k=1,n} \sum_{\ell=1,n} (\lambda_{jik} + \lambda_{ij\ell}) x_{ik} x_{j\ell} \\
& + \sum_{i=1,m} \sum_{k=1,n-1} \sum_{\ell=k+1,n} \mu_{ik\ell} x_{ik} x_{i\ell} \\
= {} & -\sum_{i=1,m} \lambda_i g_i(x) + \sum_{i=1,m} \sum_{k=1,n} \left(- \sum_{j=1,i-1} \lambda_{jik} - \sum_{j=i+1,m} \lambda_{jik} \right) x_{ik} \\
& + \sum_{i=1,m-1} \sum_{j=i+1,m} \sum_{k=1,n} \sum_{\ell=1,n} \lambda_{jik} x_{ik} x_{j\ell} \\
& + \sum_{i=1,m} \sum_{j=1,i-1} \sum_{k=1,n} \sum_{\ell=1,n} \lambda_{jik} x_{ik} x_{j\ell} \\
& + \sum_{i=1,m} \sum_{k=1,n-1} \sum_{\ell=k+1,n} \mu_{ik\ell} x_{ik} x_{i\ell} \\
= {} & \sum_{i=1,m} \lambda_i g_i(x) + \sum_{i=1,m-1} \sum_{j=i+1,m} \sum_{k=1,n} \lambda_{jik} x_{ik} g_j(x) \\
& + \sum_{i=1,m} \sum_{j=1,i-1} \sum_{k=1,n} \lambda_{jik} x_{ik} g_j(x) \\
& + \sum_{i=1,m} \sum_{k=1,n-1} \sum_{\ell=k+1,n} \mu_{ik\ell} x_{ik} x_{i\ell}. \quad \square
\end{aligned}$$

4. Computation of the best reduction

In this section, we give a theorem which states that $H_A[F(x)]$ is equal to the optimal value of a linear program and then, a corollary which shows that the posiform associated to $H_A[F(x)]$ can be easily deduced once the linear program is solved. Moreover, this posiform constitutes a reduction of $F(x)$ and therefore it is the best reduction.

Theorem 2. *Consider the linear function*

$$L_F(x,y) = \sum_{i=1,m} \sum_{k=1,n} e_{ik} x_{ik} + \sum_{i=1,m-1} \sum_{j=i+1,m} \sum_{k=1,n} \sum_{\ell=1,n} c_{ikj\ell} y_{ikj\ell}$$

obtained by rewriting $F(x)$ by introducing the new variables $y_{ikj\ell}$ in place of the product $x_{ik}x_{j\ell}$. Then, $H_A[F(x)]$ is equal to the optimal value of the following linear program:

$$(\mathrm{LP}) \quad \begin{array}{ll} \min & L_F(x,y) \\ \text{s.t.} & \displaystyle\sum_{k=1,n} x_{ik} = 1 \quad (i=1,\dots,m), \\ & \displaystyle\sum_{k=1,n} y_{ikj\ell} = x_{j\ell} \quad (i=1,\dots,m-1;\ j=i+1,\dots,m;\ \ell=1,\dots,n), \\ & \displaystyle\sum_{k=1,n} y_{j\ell ik} = x_{j\ell} \quad (j=1,\dots,m-1;\ i=j+1,\dots,m;\ \ell=1,\dots,n), \\ & y_{ikj\ell} \geqslant 0 \quad (i=1,\dots,m-1;\ j=i+1,\dots,m;\ k=1,\dots,n;\ \ell=1,\dots,n). \end{array}$$

Proof. By Lemma 1,

$$\begin{aligned} H_A[F(x)] &= \max\ \{H(F+\Gamma)\colon \Gamma \in F_0(n\cdot m, A)\} \\ &= \max_{\lambda,\mu} \left\{ H \left[F(x) + \sum_{i=1,m} \lambda_i g_i(x) + \sum_{i=1,m} \sum_{j=1,m; j\neq i} \sum_{\ell=1,n} \lambda_{ij\ell} x_{j\ell} g_i(x) \right.\right. \\ &\qquad \left.\left. + \sum_{i=1,m} \sum_{k=1,n-1} \sum_{\ell=k+1,n} \mu_{ik\ell} x_{ik} x_{i\ell} \right] \right\} \text{ by Theorem 1.} \end{aligned}$$

By expanding $(x_{j\ell} g_i(x))$, we obtain

$$\begin{aligned} H_A[F(x)] = \max_{\lambda,\mu} \left\{ H \left[F(x) + \sum_{i=1,m} \lambda_i g_i(x) \right.\right. \\ + \sum_{i=1,m} \sum_{j=1,m; j\neq i} \sum_{\ell=1,n} \lambda_{ij\ell} x_{j\ell} \left(\sum_{k=1,n} x_{ik} - 1 \right) \end{aligned}$$

$$+\sum_{i=1,m}\sum_{k=1,n-1}\sum_{\ell=k+1,n}\mu_{ik\ell}x_{ik}x_{i\ell}\Bigg]\Bigg\}$$

$$=\max_{\lambda,\mu}\Bigg\{H\Bigg[F(x)+\sum_{i=1,m}\lambda_i g_i(x)$$

$$+\sum_{i=1,m-1}\sum_{j=i+1,m}\sum_{\ell=1,n}\lambda_{ij\ell}\left(\sum_{k=1,n}x_{ik}x_{j\ell}-x_{j\ell}\right)$$

$$+\sum_{j=1,m-1}\sum_{i=j+1,m}\sum_{\ell=1,n}\lambda_{ij\ell}\left(\sum_{k=1,n}x_{j\ell}x_{ik}-x_{j\ell}\right)$$

$$+\sum_{i=1,m}\sum_{k=1,n-1}\sum_{\ell=k+1,n}\mu_{ik\ell}x_{ik}x_{i\ell}\Bigg]\Bigg\}.$$

Let C_1 be the set of constraints

$$y_{ikj\ell}\leqslant x_{ik},\quad y_{ikj\ell}\leqslant x_{j\ell},\quad 1-x_{ik}-x_{j\ell}+y_{ikj\ell}\geqslant 0,\quad y_{ikj\ell}\geqslant 0,$$

for $i=1,\ldots,m-1,\ j=i+1,\ldots,n,\ k=1,\ldots,n,\ \ell=1,\ldots,n$ and let C_2 be the set of constraints

$$y_{iki\ell}\leqslant x_{ik},\quad y_{iki\ell}\leqslant x_{i\ell},\quad 1-x_{ik}-x_{i\ell}+y_{iki\ell}\geqslant 0,\quad y_{iki\ell}\geqslant 0,$$

for $i=1,\ldots,m,\ k=1,\ldots,n-1,\ \ell=k+1,\ldots,n.$

By using the results of Hammer et al. [11] for computing the height of a q.p.B.f. (also recalled in Section 2), we obtain

$$H_A[F(x)]=\max_{\lambda,\mu}\Bigg\{\min\Bigg\{L_F(x,y)+\sum_{i=1,m}\lambda_i g_i(x)$$

$$+\sum_{i=1,m-1}\sum_{j=i+1,m}\sum_{\ell=1,n}\lambda_{ij\ell}\left(\sum_{k=1,n}y_{ikj\ell}-x_{j\ell}\right)$$

$$+\sum_{j=1,m-1}\sum_{i=j+1,m}\sum_{\ell=1,n}\lambda_{ij\ell}\left(\sum_{k=1,n}y_{j\ell ik}-x_{j\ell}\right)$$

$$+\sum_{i=1,m}\sum_{k=1,n-1}\sum_{\ell=k+1,n}\mu_{ik\ell}y_{iki\ell}:$$

$$x \text{ and } y \text{ subject to } C_1 \text{ and } C_2\Bigg\}\Bigg\}.$$

Therefore, $H_A[F(x)]$ can be considered as the optimal value of the Lagrangean dual problem obtained from the following continuous linear program:

$$\begin{aligned}
&\min \quad L_F(x,y)\\
&\text{s.t.} \quad g_i(x)=0 \quad (i=1,\dots,m), && \text{(a)}\\
&\sum_{k=1,n} y_{ikj\ell}=x_{j\ell} \quad (i=1,\dots,m-1;\ j=i+1,\dots,m;\ \ell=1,\dots,n), && \text{(b)}\\
&\sum_{k=1,n} y_{j\ell ik}=x_{j\ell} \quad (j=1,\dots,m-1;\ i=j+1,\dots,m;\ \ell=1,\dots,n), && \text{(b}'\text{)}\\
&y_{iki\ell}=0 \quad (i=1,\dots,m;\ k=1,\dots,n-1;\ \ell=k+1,\dots,n)\\
&\quad (\mathrm{C}_1) \text{ and } (\mathrm{C}_2) && \text{(c)}
\end{aligned}$$

by "dualizing" the constraints (a)–(c). Indeed, it is easy to verify that the Lagrangean function

$$\begin{aligned}
&L_F(x,y)+\sum_{i=1,m}\lambda_i g_i(x)\\
&\quad+\sum_{i=1,m-1}\ \sum_{j=i+1,m}\ \sum_{\ell=1,n}\lambda_{ij\ell}\left(\sum_{k=1,n} y_{ikj\ell}-x_{j\ell}\right)\\
&\quad+\sum_{j=1,m-1}\ \sum_{i=j+1,m}\ \sum_{\ell=1,n}\lambda_{ij\ell}\left(\sum_{k=1,n} y_{j\ell ik}-x_{j\ell}\right)\\
&\quad+\sum_{i=1,m}\ \sum_{k=1,n-1}\ \sum_{\ell=k+1,n}\mu_{ik\ell}\, y_{iki\ell}
\end{aligned}$$

is obtained by assigning the Lagrangean multipliers λ_i $(i=1,\dots,m)$ to the constraints (a), the Lagrangean multipliers $\lambda_{ij\ell}$ $(i=1,\dots,m-1;\ j=i+1,\dots,m;\ \ell=1,\dots,n)$ to the constraints (b), the Lagrangean multipliers $\lambda_{ij\ell}$ $(j=1,\dots,m-1;\ i=j+1,\dots,m;\ \ell=1,\dots,n)$ to the constraints (b$'$) and the Lagrangean multipliers $\mu_{ik\ell}$ $(i=1,\dots,m;\ k=1,\dots,n-1;\ \ell=k+1,\dots,n)$ to the constraints (c). Therefore, $H_A[F(x)]$ is equal to the optimal value of this linear program. To prove the theorem, we now need to prove that the constraints ((c), (C_1) and (C_2) apart from $(y_{ikj\ell}\geqslant 0)$) are useless. This is easy to check except for the following ones: $1-x_{ik}-x_{j\ell}+y_{ikj\ell}\geqslant 0$ $(i=1,\dots,m-1;\ j=i+1,\dots,n;\ k=1,\dots,n;\ \ell=1,\dots,n)$ for which one can write by using constraints (a)–(b$'$): $1-x_{ik}-x_{j\ell}+y_{ikj\ell}=\sum_{r=1,n;r\neq k} x_{ir}-x_{j\ell}+y_{ikj\ell}=\sum_{r=1,n;r\neq k} x_{ir}-\sum_{r=1,n;r\neq k} y_{irj\ell}-y_{ikj\ell}+y_{ikj\ell}=\sum_{r=1,n;r\neq k}(x_{ir}-y_{irj\ell})$ and this last quantity is nonnegative. □

Corollary 1. *The best reduction of $F(x)$ is*

$$F'(x)=\sum_{i=1,m}\ \sum_{k=1,n}\delta_{ik}x_{ik}+\sum_{i=1,m-1}\ \sum_{j=i+1,m}\ \sum_{k=1,n}\ \sum_{\ell=1,n}\delta_{ikj\ell}x_{ik}x_{j\ell},$$

where δ_{ik} and $\delta_{ikj\ell}$ are the (positive) reduced costs of the variables x_{ik} and $y_{ikj\ell}$, respectively, obtained by solving (LP) *with the simplex algorithm.*

Proof. The simplex algorithm gives, at the optimum of the linear program of Theorem 2, the following equality: $L_F(x,y)=H_A[F(x)]+\sum_{i=1,m}\sum_{k=1,n}\delta_{ik}x_{ik}+\sum_{i=1,m-1}\sum_{j=i+1,m}\sum_{k=1,n}\sum_{\ell=1,n}\delta_{ikj\ell}y_{ikj\ell}$, where δ_{ik} and $\delta_{ikj\ell}$ are the (positive) reduced costs of the variables x_{ik} and $y_{ikj\ell}$, respectively. It is easy to derive from this equality that $F(x)=H_A[F(x)]+\sum_{i=1,m}\sum_{k=1,n}\delta_{ik}x_{ik}+\sum_{i=1,m-1}\sum_{j=i+1,m}\sum_{k=1,n}\sum_{\ell=1,n}\delta_{ikj\ell}x_{ik}x_{j\ell}$. □

Remark. (i) (LP) involves $mn+(m^2n^2-mn^2)/2$ variables and $(m^2n-mn+m)$ constraints (plus the nonnegativity constraints of the variables). If we do not want to solve (LP) until the optimum, we can use the fact that each feasible solution of the dual of (LP) gives a reduction of $F(x)$.

(ii) The result of Theorem 2, which gives the best reduction of the semi-assignment problem, can be easily strengthened to the following slightly more general problem:

$$\begin{aligned} \min\ & \sum_{i=1,n} q_i x_i + \sum_{i=1,n-1}\ \sum_{j=i+1,n} q_{ij}x_i x_j \\ \text{s.t.}\ & g_t(x)=0 \quad (t=1,2,\dots,T), \\ & x\in\{0,1\}^n, \end{aligned}$$

where q_i $(i=1,\dots,n)$ and $q_{i,j}$ $(i=1,\dots,n-1;\ j=i+1,\dots,n)$ are real numbers; q_{ij} is equal to zero whenever $i\geqslant j$, $(I_1,I_2,\dots,I_T)$ is a partition of $\{1,2,\dots,n\}$ and $g_t(x)=\sum_{i\in I_t}x_i-1$.

5. Discussion and some experimental results

We proved in the first sections of this paper that the best possible reduction of (QSAP) could be obtained by solving a certain linearization of the problem. This linearization itself is not new. In Section 5.1, we study it with reference to the literature. On the other hand, one can imagine many ways of calculating a reduction of (QSAP) within the meaning of Definition 1. We present in Section 5.2 some of these reductions and show that there are instances where the reduction corresponding to Theorem 2 and Corollary 1 is strictly better than the others. Section 5.3 is devoted to experimental results which show that this last reduction approach is indeed useful in practice.

5.1. About the linearization technique yielding to problem (LP)

In the previous sections, we show that an optimal reduction of (QSAP) can be obtained by solving the linear program (LP). Observing this program, one can easily check that it is also induced by a well-known linearization technique initially proposed in [3] for the general linearly constrained 0–1 quadratic problem. Earlier, the authors of [9] introduced a similar linearization technique for the Quadratic Assignment Problem (QAP) and several other authors have considered this linearization technique from the quality of the obtained lower bound point of view. In [2,14], it is applied to the (QAP) and the lower bounds computed by solving the corresponding linear programs (up to $n = 30$ in [14]) are showed to be better than most other lower bounds available in the literature. More recently, the authors of [12,13] carried out an extensive polyhedral study for (QAP) and presented this study as an efficient way to sharpen the previous bounds by removing redundant equalities from the obtained linear program and adding facets. In [6,12] are also reported preliminary computational results which show that solving the linear problems can be done more quickly if the sparse characteristic of the objective function is taken into account. They point out this last observation as a promising axe for future research.

5.2. The best reduction of Section 4 is strictly better than some other classical reductions

There are standard techniques to reduce (QSAP) (in the meaning of Definition 1). We present below three of them together with the reduction of Section 4 that we call Reduction 1. Of course, the results of the previous sections prove that Reduction 1 is better than the three other ones.

Reduction 1. $\min\{H_A[F(x)] + F'(x)\colon x \in A\}$, where $H_A[F(x)]$ is the optimal value of (LP) and $F'(x)$ is directly obtained from the reduced costs corresponding to the optimal solution of (LP) (see Corollary 1).

Reduction 2. In this reduction the technique consists to write the objective function F as $H[F(x)] + \phi(x,\bar{x})$ where ϕ is a quadratic posiform. $H[F(x)]$ and the corresponding posiform ϕ can be obtained by solving the linear program (LP2) (see [11]):

$$
\text{(LP2)}\quad
\begin{array}{ll}
\min & L_F(x,y) \\
\text{s.t.} & \left.\begin{array}{l}
y_{ikj\ell} \leqslant x_{ik}, \\
y_{ikj\ell} \leqslant x_{j\ell} \\
1 - x_{ik} - x_{j\ell} + y_{ikj\ell} \geqslant 0, \\
y_{ikj\ell} \geqslant 0.
\end{array}\right\}
\begin{array}{l}
(i = 1,\ldots,m-1;\ j = i+1,\ldots,m; \\
k = 1,\ldots,n;\ \ell = 1,\ldots,n),
\end{array}
\end{array}
$$

Moreover, using the constraints of (QSAP) enables us to express a complemented variable only in function of direct variables and then to write $\phi(x,\bar{x}) = F_2(x)$ for all $x \in A$ where F_2 is a quadratic 0–1 function without constant term. So the reduced (QSAP) is $\min\{H[F(x)] + F_2(x)\colon x \in A\}$.

Reduction 3. Another reduction technique consists to consider the linear program (LP3)

$$
\text{(LP3)}\quad
\begin{array}{ll}
\min & L_F(x,y) \\
\text{s.t.} & \displaystyle\sum_{k=1,n} x_{ik} = 1 \quad (i=1,\ldots,m), \\
& \left.\begin{array}{l}
y_{ikj\ell} \leqslant x_{ik}, \\
y_{ikj\ell} \leqslant x_{j\ell}, \\
1 - x_{ik} - x_{j\ell} + y_{ikj\ell} \geqslant 0 \\
y_{ikj\ell} \geqslant 0.
\end{array}\right\}
\begin{array}{l}
(i=1,\ldots,m-1;\ j=i+1,\ldots,m; \\
k=1,\ldots,n;\ \ell=1,\ldots,n),
\end{array}
\end{array}
$$

The reduced function is derived from the solution of (LP3) in the same way as in Reduction 2.

Reduction 4. By a classical penalty function method, (QSAP) can be converted to the unconstrained problem $\min\{G(x) = F(x) + M\sum_{i=1,m}[g_i(x)]^2 \colon x \in \{0,1\}^{m\times n}\}$, where M is a sufficiently large positive number. Denote by $L_G(x)$ the linear function obtained by rewriting $G(x)$ by introducing the new variables $y_{ikj\ell}$ in place of the product $x_{ik}x_{j\ell}$. As in Reduction 2 one can compute $H(G)$ and a reduction F_4 such that $G(x) = H(G) + F_4(x)$ for all $x \in A$ by solving the following linear program (LP4):

$$
\text{(LP4)}\quad
\begin{array}{ll}
\min & L_G(x,y) \\
\text{s.t.} & \left.\begin{array}{l}
y_{ikj\ell} \leqslant x_{ik}, \\
y_{ikj\ell} \leqslant x_{j\ell}, \\
1 - x_{ik} - x_{j\ell} + y_{ikj\ell} \geqslant 0 \\
y_{ikj\ell} \geqslant 0.
\end{array}\right\}
\begin{array}{l}
(i=1,\ldots,m-1;\ j=i+1,\ldots,m; \\
\quad k=1,\ldots,n;\ \ell=1,\ldots,n), \\
(i=1,\ldots,m;\ j=i;\ k=1,\ldots,n; \\
\quad \ell=1,\ldots,n;\ \ell \neq k),
\end{array}
\end{array}
$$

Using Lemma 1, one can verify that Reduction 1 is better than Reduction 4 because $G(x)$ is equal to $F(x)$ plus a particular function equal to zero for all $x \in A$.

5.2.1. Computation of the four previous reductions on an example

Now consider the following quadratic semi-assignment problem Π with $m=3$ and $n=3$:

$$
\begin{aligned}
\min F(x) = {} & -3x_{11} - 3x_{12} - 5x_{21} + 4x_{22} - 4x_{33} + 2x_{11}x_{21} + 5x_{12}x_{21} + 6x_{12}x_{33} \\
& + 5x_{21}x_{33}
\end{aligned}
$$

s.t. $x \in A$.

The four reduction techniques lead to four reduced (QSAP) problems, which show that, on this instance, Reduction 1 is strictly better than the three others.

Reduction 1.

$$\begin{aligned}\min \quad & (\mathbf{-7}) + x_{21} + 4x_{22} + 3x_{12}x_{21} + 6x_{12}x_{33} + x_{21}x_{33} + x_{13}x_{21} + 3x_{13}x_{22} + 3x_{13}x_{23} \\ & \quad + 4x_{22}x_{31} + 4x_{23}x_{31} + 4x_{22}x_{32} + 4x_{23}x_{32} \\ \text{s.t.} \quad & x \in A.\end{aligned}$$

Reduction 2.

$$\begin{aligned}\min \quad & (\mathbf{-8}) + 4x_{22} + x_{11}x_{22} + x_{11}x_{23} + 2x_{11}x_{31} + 2x_{11}x_{32} + 5x_{12}x_{21} + 3x_{12}x_{22} \\ & \quad + 3x_{12}x_{23} + 4x_{12}x_{33} + x_{13}x_{21} + 4x_{13}x_{22} + 4x_{13}x_{23} + 2x_{13}x_{31} + 2x_{13}x_{32} \\ & \quad + 3x_{21}x_{33} + 2x_{22}x_{31} + 2x_{22}x_{32} + 2x_{23}x_{31} + 2x_{23}x_{32} \\ \text{s.t.} \quad & x \in A.\end{aligned}$$

Reduction 3.

$$\begin{aligned}\min \quad & (\mathbf{-7.5}) + 4x_{22} + x_{13} + 0.5x_{11}x_{22} + 0.5x_{11}x_{23} + 1.5x_{11}x_{31} + 4.5x_{12}x_{21} + 2x_{12}x_{22} \\ & \quad + 2x_{12}x_{23} + 4.5x_{12}x_{33} + 2.5x_{13}x_{22} + 2.5x_{13}x_{23} + 1.5x_{13}x_{31} + 3.5x_{13}x_{32} \\ & \quad + 2.5x_{21}x_{33} + 2.5x_{22}x_{31} + 2.5x_{22}x_{32} + 2.5x_{23}x_{31} + 2.5x_{23}x_{32} \\ \text{s.t.} \quad & x \in A.\end{aligned}$$

Reduction 4 ($M = 20$)**.**

$$\begin{aligned}\min \quad & (\mathbf{-35.5}) + 8x_{11} + 12x_{12} + 9x_{13} + 9x_{21} + 11x_{22} + 7x_{23} + 6.5x_{31} + 6.5x_{32} \\ & \quad + 13.5x_{33} + 6x_{11}x_{31} + 6x_{11}x_{32} + 5x_{12}x_{21} + 2x_{12}x_{22} + 2x_{12}x_{23} + 2x_{13}x_{22} \\ & \quad + 2x_{13}x_{23} + 6x_{13}x_{31} + 6x_{13}x_{32} + 5x_{22}x_{31} + 5x_{22}x_{32} + 5x_{23}x_{31} + 5x_{23}x_{32} \\ \text{s.t.} \quad & x \in A.\end{aligned}$$

Note that the optimal solution of Π is $x_{11} = x_{23} = x_{33} = 1$ with an objective value equal to -7.

5.3. A practical use of the best reduction

The main result of this work is of theoretical nature. It shows that Reduction 1 is the best possible reduction of the problem as defined in Section 1. From a practical point of view, one can wonder whether this reduction can be exploited for the resolution of (QSAP) since, as we indicated in Section 1, after the problem has been reduced we again have a problem (QSAP) to which one can apply various solution techniques. We give below experimental results which prove that, at least for a standard technique of resolution of (QSAP), it is preferable to carry out as a presolving the reduction of the problem. More precisely, we compared the two following methods for solving (QSAP):

Method 1. Resolution of (QSAP) by solving the classical 0–1 linear program (MIP3) obtained by adding to problem (LP3) of Reduction 3, the integrality condition on variables x.

Method 2. Computation of the reduced function F' by solving the continuous linear program (LP) then resolution of the reduced problem by the program (MIP3r) obtained by substituting $L_{F'}(x,y)$ to $L_F(x,y)$ in program (MIP3).

The instances of QSAP are randomly generated in the following way: we choose the parameters m and n who define the problem dimension and then the coefficients e_{ik} and $c_{ikj\ell}$ are randomly selected from an interval $[-50,50]$. The different linear programs, continuous or integer, are solved using CPLEX 6.5 on a Sun Solaris workstation. Table 1 displays the computational results on these instances of (QSAP). Column 1 gives the value of n and column 2, the value of m. Columns 3 and 4 concern the resolution of (QSAP) by the first method and columns 5–8, the resolution of (QSAP) by the second one. Column 3 gives the CPU time required to solve the 0–1 linear program (MIP3) associated with the considered instance and column 4, the number of nodes in the corresponding search tree. Column 5 gives the CPU time required to carry out the reduction, i.e. to solve the continuous linear program (LP), column 6, the CPU time required for solving the reduced problem (MIP3r) and column 7, the sum of these two times. Column 8 gives the number of nodes in the resolution of the reduced problem (MIP3r). Lastly, column 9 shows the obtained speed up, i.e. $\lfloor$CPU time Method 1/total CPU time Method 2$\rfloor$.

The results presented in Table 1 show that, on the considered instances, the resolution of (QSAP) by Method 2 is much faster than by Method 1 (between 2 and 50 times faster). It is also noted that the computing time necessary to reduce the problem (column 5) is very weak compared to the total computing time. Lastly, neither Method 1 nor Method 2 allow the larger instances to be solved but it is seen that the problem can still be reduced in a reasonable computing time although the number of variables and constraints of the linear program associated with the reduction are large. For example, for $n=7$ and $m=35$, the program LP comprises 29 400 variables and 8365 constraints.

Table 1
Comparison of the resolution of (QSAP) without a preliminary reduction (Method 1) and after a preliminary reduction (Method 2)

1 2 n m	3 Method 1 CPU time (s)	4 Method 1 no. nodes	5 Method 2 CPU time reduction (s)	6 Method 2 CPU time reduced problem (s)	7 Method 2 Total CPU time (s)	8 Method 2 no. nodes	9 Speed up
3 5	0.25	12	0.01	0.07	0.08	2	3
3 5	0.19	4	0.03	0.05	0.08	4	2
3 5	0.24	2	0.03	0.00	0.03	0	8
3 10	9.10	120	0.19	0.57	0.76	15	11
3 10	8.60	77	0.18	0.71	0.89	31	9
3 10	9.70	168	0.20	0.60	0.80	19	12
4 12	540	3530	0.95	22	22.95	1247	23
4 12	320	1458	0.88	8.30	9.18	390	34
4 12	780	2634	0.89	14	14.89	701	52
2 15	7.70	265	0.34	0.34	0.68	12	11
2 15	7.80	318	0.36	0.32	0.68	11	11
2 15	9.00	328	0.37	0.46	0.83	32	10
5 15	>6000	—[a]	4.20	2300	2304.2	62599	x[b]
5 15	>6000	—[a]	3.50	3300	3303.5	82221	x[b]
5 15	>6000	—[a]	3.60	2500	2503.6	46371	x[b]
4 18	>6000	—[a]	4.80	1600	1604.8	43026	x[b]
4 18	>6000	—[a]	4.70	2100	2104.7	49611	x[b]
4 18	>6000	—[a]	4.80	1300	1304.8	34287	x[b]
4 20			8.20				
4 25			21				
4 30			50				
5 20			15				
5 35			300				
6 30			220				
7 35			910				

[a]No optimal solution was found after 6000 s of CPU time.
[b]Speed up has no sense for these instances.

6. Conclusion

Our objective in this paper is not to propose a linear programming bound for (QSAP) but to find a method for computing its best reduction. We prove that the best reduction of (QSAP) can be obtained by solving the linear program (LP) of Theorem 2. This linear program can be viewed as the continuous relaxation of a certain linearization of the semi-assignment problem. Several authors considered this linearization technique from a lower bound point of view. For example, it was introduced in [9] for the Quadratic Assignment Problem (QAP) and in [3] for the general linearly constrained zero-one quadratic problem. More recently, the authors of [14] used this linearization in order to compute a lower bound for the (QAP). They showed in an exhaustive experimentation that the obtained linear program can be solved efficiently by an interior point code and that, on 87% of the instances, the computed bound is the best known lower bound. It is straightforward that this linearization leads to a reduction of the (QAP). It would be interesting to show it is the best one. In order to prove this result, the main difficulty is to characterize the quadratic pseudo-Boolean functions equal to zero for all assignments.

Ref. [11] shows the equivalence of linearization presented in [15] with the roof dual and the determination of the height for unconstrained 0–1 quadratic optimization. In the case of (QSAP), we proved in this paper that the linear program for finding the height (extended to the constrained case) can be interpreted as a certain linearization of the problem, which moreover provides a best reduction. The connection with roofs is less evident and would be an interesting topic for future work as suggested by the referees.

References

[1] W.P. Adams, A. Billionnet, A. Sutter, Unconstrained 0–1 optimization and Lagrangean relaxation, Disc. Appl. Math. 29 (1990) 131–142.

[2] W.P. Adams, T.A. Johnson, Improved linear programming-based lower bounds for the quadratic assignment problem. Proceedings of the DIMACS Workshop on Quadratic Assignment Problems DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society 1994, Vol. 16, pp. 43–75.

[3] W.P. Adams, H.D. Sherali, A tight linearization and an algorithm for zero–one quadratic programming problems, Management Sci. 32 (10) (1986) 1274–1290.

[4] A. Billionnet, M.-C. Costa A. Sutter, An efficient algorithm for a task allocation problem, J. Assoc. Comput. Mach. 39 (3) (1992) 502–518.

[5] A. Billionnet, A. Faye, A lower bound for a constrained quadratic 0–1 minimization problem, Disc. Appl. Math. 74 (1997) 135–146.

[6] M. Elf, LP-basierte Schranken fur quadratsche Zuordnungsprobleme mit dunner Zielfunktion, Master's Thesis, Universitat zu Koln, 1999.

[7] S. Elloumi, Contribution à la résolution des programmes non linéaires en variables 0–1, application aux problèmes de placement de tâches dans les systèmes distribués, Thèse de Doctorat en informatique, Conservatoire National des Arts et Métiers, Paris, September 1991.

[8] A. Faye, Programmation quadratique en variables bivalentes sous contraintes linéaires, Application au placement de tâches dans les systèmes distribués et à la partition de graphes, Thèse de Doctorat en informatique, Conservatoire National des Arts et Métiers, Paris, January 1994.

[9] A.M. Frieze, J. Yadegar, On the Quadratic Assignment Problem, Disc. Appl. Math. 5 (1983) 89–98.

[10] G. Gallo, B. Simeone, Optimal grouping of researchers into departments, Ricerca Operativa 57 (1991) 45–69.

[11] P.L. Hammer, P. Hansen, B. Simeone, Roof duality, complementation and persistency in quadratic 0–1 optimization, Math. Program. 28 (1984) 121–155.

[12] V. Kaibel, Polyhedral Methods for the QAP. A chapter written for a book on nonlinear assignment problems, edited by Panos Pardalos and Leonidas Pitsoulis, Kluwer Academic Publishers, Dordrecht, 1999. http://www.math.TU-Berlin.de/∼kaibel/pub.html, to appear.

[13] M.W. Padberg, M.P. Rijal, Location, Scheduling, Design and Integer Programming, Kluwer Academic Publishers, Dordrecht, 1996.

[14] M.G.C. Resende, K.G. Ramakrishnan, Z. Drezner, Computing lower bounds for the Quadratic assignment problem with an interior point algorithm for linear programming, Oper. Res. 43 (5) (1995) 781–791.

[15] J. Rhys, A selection problem of shared fixed costs and networks, Manag. Sci. 17 (1970) 200–207.

[16] A. Sutter, Programmation non linéaire en variables 0–1, application à des problèmes de placement de tâches dans des systèmes distribués, Thèse de Doctorat en informatique, Conservatoire National des Arts et Métiers, Paris, June 1989.